Express Mail No.   EL633839553US

METHOD TO DYNAMICALLY PERFORM
DOCUMENT LAYOUT FUNCTIONS

Background of the Invention

This invention pertains generally to document formatting, and more specifically to a method of formatting a document printed from a software program in a selected format and at a selected resolution.

Networks today utilize a variety of different operating systems, both on the server side and on the client side. A few common examples of such operating systems are the Windows family of operating systems, Novell, Unix, Linux, as well as various mainframe systems. The many operating systems have different capabilities and interface differently with printer drivers. Consequently, certain user functionality that is available on one system may not be available on other systems.

Generally, a user is able to submit document to a printer for printing in a variety of different formats. Most software applications permit a user to format a document so that it prints in booklet format (2-up) or as any multiple n of pages on a single printed page (n-up) where n is an integer greater than 1. In some circumstances, however, operating systems, software programs and printer drivers are unable to interact properly in order to create an n-up printing format. In such cases, printer drivers generally format the print job and send it to a print controller. As such, when a print job is sent from a print driver to a print controller, it is formatted so that the print controller need not perform any further processing in order to produce a correctly formatted printed document.

Generally, in order to send preformatted pages to a print controller, a print driver must perform all required formatting whenever a particular software program is unable to format the pages as desired. For example, if the desired format is booklet form, the print driver must shrink the output of the software package so that each page received from the software program fits on one half of a booklet format page. The shrinking of the pages alters the image of a particular page and results in data loss. However, some printers have a native ability to generate formatted

documents after receiving unformatted data. Using the same booklet example, a print job would be formatted into booklet form after being sent to the printer and prior to printing.

It is preferable to allow a printer, rather than a print driver, to perform formatting because it decreases both network traffic and potential compatibility issues. Because each driver is designed to interface with a particular operating system, drivers tend to be operating system specific. In order to build cross-platform formatting capability into printer drivers to operate a single printer, formatting capability must be built into each operating system specific printer driver. However, while each driver is specific to the particular operating system with which it interfaces, each printer will work with a variety of printer drivers, allowing for cross-platform functionality.

It would therefore be preferable if a print driver were capable of first determining printer capability, and then using that information to determine where document formatting will occur.

Summary of the Invention

According to the present invention, there is provided a method of formatting a document to be printed from a software program in a selected format and at a selected resolution. The first step of the method involves determining if the software program from which the user is printing is capable of creating the document in the selected format and at the selected resolution. If the software program is not capable of creating the document in the selected format and at the selected resolution, an appropriate output resolution for at least a portion of the document is determined based upon the selected format and the selected resolution. The resolution of at least a portion of the document is altered by the software program prior to the generation of the document for printing by the software program. The resulting generated document, which is in a format and resolution different than that desired, is then assembled such that when the altered resolution document is printed at the selected resolution, the format of the printed document matches the selected format. The document is then printed at the selected resolution.

Description of the Figures

Figure 1 is a flow chart representing the general process of formatting a document for printing according to the present invention;

Figure 2 is a flow chart according to the present invention representing the process of formatting a document for booklet format, and further representing the process involved in the formatting of a document by the driver;

Figure 3 is a flow chart according to the present invention representing the process of formatting a document for booklet format by a controller;

Figure 4 is a representation of a page in booklet or 2-up format;

Figure 5 is a representation of a page in 4-up format; and

Figure 6 is a representation of a network.

Detailed Description of the Invention

The present invention involves a method and system for formatting documents in n-up format when a software program is incapable of creating an n-up format document. The invention applies to any system comprising a computer running on any operating system that uses at least one printer driver and is connected to at least one printer. In a preferred embodiment, the computer and printer are both connected to a network. The network suitably runs on any network operating system ("NOS") or mainframe system and suitably supports a myriad of software packages. Preferably the present invention is employed in a system where a user of personal computer ("PC") that issues a "print" command to create a hard copy of a document. The hard copy is printed from a printer, which is suitably a multifunction peripheral ("MFP") device such as a combined copier/printer/fax device. Preferably, the system has a raster image processor ("RIP"). In one preferred embodiment, the user selects a "print" command from a menu of command options, such as those used with a graphical user interface operating system. However, it should be understood that the present invention is not limited in this manner and could easily be adapted to systems employing a print or other type function, all without departing from the present invention.

Turning to **Figure 1**, a flow diagram of a method for formatting a document for printing in a selected format and at a selected resolution is disclosed. The basic flow commences at manual operation **10** when a user invokes a print command from a software program. The

3

software program is not limited to any particular application software program. The software program is suitably any program in an office suite, a word processing package, or even the operating system itself, e.g. printing from a DOS environment. When a user invokes a print command, the user is prompted to select a print layout and print resolution. In a preferred embodiment, a printer driver on the computer activates a printer dialog box that pops up to enable the user to select various print options including layout and resolution. Upon selecting the desired print layout and print resolution **10**, the user-selected options are sent to a printer driver, which receives the selected layout and resolution **12**.

A determination is then made at decision block **14** if the software program is capable of creating a printable document in the desired format and at the desired resolution. The determination is suitably accomplished by an examination of the program output. Preferably, the determination is made by a printer driver. A positive determination at decision block **14** causes progression to block **16**. At block **16**, the software program formats the document for printing and flow progresses to block **26** where the document is sent in formatted form to the controller for printing.

A negative determination at decision block **14** causes progression to block **18**. At this point, the software program is instructed to change the print resolution. Preferably, this instruction to change the resolution of the output of the software program is accomplished by the printer driver. In a presently preferred embodiment, the printer driver "tricks" the software program into printing at a resolution that is lower different than the resolution selected by the user. When the print or output resolution is changed, it is altered so that when the document is printed at the resolution selected by the user, the document is formatted in a manner consistent with the layout selected by the user.

For example, if a user selects booklet layout (2-up) document, the software program is instructed to generate a document that is one half the resolution of the user-selected resolution. Furthermore, if the user selects a 4-up layout, the software program is instructed to generate a document that is one fourth the resolution of the selected resolution. Therefore, in one embodiment of the present invention, the software program is suitably instructed to generate a document having a resolution equal to the user-selected resolution divided by the number of pages to be displayed per page in the user-selected output. It should be noted that while this general formula applies to n-up formatted documents, the present invention is not limited to n-up

documents. The process of instructing a software program to generate a document at a lower resolution is applicable to any format document where a user selects a layout that requires the shrinking of an image in order to accommodate the selected format.

Following the changing of the print resolution of the software program output, a determination is made at decision block **20** whether the printer controller is capable of generating the layout selected by the user from the altered resolution software program output. A positive determination at decision block **20** causes progression to process **24** where the software program output is forwarded to the controller without further formatting. Flow then progresses to block **28** where the controller formats the document for printing, after which the document is printed and the and the system proceeds to stop at termination block **30**.

A negative determination at decision block **20** causes progression to process **22** where the driver, rather than the controller, formats the document for printing. Flow then progresses to process **26** where the formatted document is sent to the controller. After the controller receives the formatted document, the document is printed and the system proceeds to stop at termination block **30**.

Turning next to **Figure 2**, a flow diagram of a method for formatting a document for printing in a booklet format and at a selected resolution is disclosed. Again, the basic flow commences at manual operation **10** when a user invokes a print command from a software program. The software program is not limited to any particular application software program. The software program is suitably any program, such as an office suite, a word processing package, or even the operating system software itself, e.g. printing from a DOS environment. When a user invokes a print command, the user is prompted to select a print layout and print resolution. Upon selecting the desired booklet layout and choosing a desired print resolution, the user-selected options are sent to a printer driver, which receives the selected layout and resolution.

Flow progresses to decision block **34** where a determination is made whether the software program from which a user invoked the print dialog is capable of formatting a document in booklet format. In the presently preferred embodiment, the determination is made by the printer driver through examination of the software program output. A positive determination at decision block **14** causes progression to process **36**, where the software program assembles the printed

pages in booklet format in proper order. Flow then progresses to processes **54** where the assembled formatted pages are sent to the controller for printing.

A negative determination at decision block **34** causes progression to process **38** where the software program is instructed to generate a standard format output at a resolution different than that selected by the user in manual operation **32**. In the presently preferred embodiment, the software program is instructed to print at a resolution that is one half of that selected by the user. For example, if a user selected booklet format layout and a print resolution of 600 dots per inch (dpi), the software program is instructed to create a 300 dpi document in standard, non-booklet printing format.

Flow then progresses to decision block **40** where a determination is made whether the printer controller is capable of creating a booklet format output at the resolution selected by the user in manual operation **32**. A positive determination at decision block **40** causes progression to process **42** where the controller is set to duplex mode printing. In the presently preferred embodiment, the printer driver causes the controller to change from simplex mode to duplex mode printing. Flow progresses to process **44** where the print job is then sent to the controller for both formatting and printing. The controller then assembles the print pages in booklet format in proper order during process **46**. Following assembly, flow progresses to process **56** where the print data is generated by the controller. In a presently preferred embodiment, the controller receives a document formatted for a resolution that is one half that of the resolution selected by the user and prints the document at the resolution selected by the user. Consequently, the print data sent to the printer will print at one half the size of that generated by the software program, thus allowing for a booklet page layout as two pages will fit on one printed page. The document is then printed by the printer and the system proceeds to stop at termination block **58**.

A negative determination at decision block **40** causes progression to process **48** where the booklet generation capabilities of the driver are enabled. The driver then clears the booklet command in the print data generated by the software program during process **50**. Flow progresses to process **52** where the driver assembles the print pages in booklet format and in proper order. The print data is then sent to the controller for printing in process **54**. At process **54**, the print data is pre-assembled by either the software program in process **36** or the printer driver in process **52**. Flow then progresses to process **56** where the controller generates the print

data at the user-selected resolution. The document is then printed by the printer and the system proceeds to stop at termination block **58**.

Turning now to **Figures 3**, a flow diagram is disclosed for generating a booklet format document as shown in **Figure 4**. The flow commences upon receipt of print job data **60**. The print job data is received from either a software program or a print driver, depending on whether the print driver or controller is generating the booklet format. If the driver generates the booklet format, it will receive a print job from the software program. However, if the printer controller generates the booklet format, it will receive a print job from the driver.

Flow progresses to process **62** where booklet commands and parameters are parsed. The print data including booklet commands and parameters is send to the raster image processor ("RIP") in process **64**. Flow progresses to process **66** where the RIP takes the print data and booklet commands and creates a bitmap image of each page to be printed in booklet format. The bitmap images are created at one half the user-selected print resolution. Flow progresses to process **68** where the bitmap images of pages created by the RIP are collected and reassembled for printing. In the reassembly process, the bitmap images for each page are placed on a canvas for printing at the user selected resolution. Because the bitmap images are one half the resolution of the canvas for printing, the images actual printing size is one half of the size of the output by the software program. Therefore, the pages are suitably assembled so that two bitmap images are placed on one new page or canvas for printing.

Once the bitmap images are created for placement on one half of a page for printing, flow progresses to process **70** where gutter space **84**, creep **86** and margins **88** are applied to each new page upon which there are placed bitmap images **80** and **82** of pages received by the RIP. The gutter space is defined as the distance between pages **80** and **82**. The creep **86** is the change in gutter space **84** required as the number of pages in a document grows. The reason for the increase in gutter space **84** is that paper has a defined thickness, depending on the type of stock used. As growing numbers of pages are folded in half to create a booklet, the outer sheets seem shorter and shorter. Once the pages are evenly cut, margins **84** shift noticeably between the middle and the beginning and ending of the booklet. The creep **86** increases the gutter **84** of the pages **80** and **82** so that text is correctly positioned on each imposed sheet.

7

Following the application of gutter space **84**, creep **86** and margins **88** to the new booklet format page, flow progresses to termination block **72** where the document is printed and the system proceeds to stop.

Turning now to **Figure 5**, a 4-up page layout is shown. A document of the type shown in Figure **5** is suitably created according to the present invention by generating a document from a software program having a resolution of one fourth that of the selected print resolution. When the RIP generates bitmap images of print data, the bitmap images are one fourth the size of a page, rather than one half as they are when a booklet format is selected. The RIP suitably collects the bitmap images and reassembles them onto a new page, where four images are assembled per page. While **Figure 5** depicts an across, then down order of pages, a down, then across order of pages may also be created according the present invention. Furthermore, when creating a 4-up document, the RIP need not adjust creep or gutter space.

Turning now to **Figure 6,** there is provided a diagram of a network system for practicing the present invention. The network **100** is illustrative of a LAN or WAN environment in which a preferred embodiment is provided. Connected to a network **100** are a client machine **102** and a printer **104**. The network **100** is suitably any network and is suitably comprised of physical and transport layers such as illustrated by a myriad of conventional data transport mechanisms such as Ethernet, Token-Ring™, 802.11(b), or other wire-based or wireless data communication mechanisms as will be apparent to one of ordinary skill in the art. The client machine is suitably any type of computer, but is preferably a PC running an on a Windows, Unix, Macintosh, or Linux based operating system. The client machine **102** is equipped with a software program **112**, a printer driver **114**, and a network connector **116**. It should be noted that the printer driver **116** is suitably loaded on the client machine or on a network server.

The printer **104** is suitably any networked printer. However, it should be noted that while the preferred embodiment of the present invention is a network system having multiple client machines and at least one printer, the present invention includes systems having printers and personal computers connected directly to one another through a port such as a USB or a LPT port. The printer **104** is equipped with a printer controller **106**, a network connector **108** and internal storage **110**. The printer internal storage **110** is suitably a hard disk or random access memory ("RAM").

8

When a user of the client machine **102** invokes a "print" command from a software program **112**, the printer driver **114** is activated. The printer driver **114** causes a selection window to be displayed on a computer monitor connected to the client machine **102**. The user then selects a layout and a print resolution. If the software program **112** is capable of generating

5   a document in the selected layout and resolution, the printer driver **114** sends the data generated by the software program directly to the printer **104**. However, if the software program **112** is not capable of generating the selected layout and resolution, the printer driver **114** instructs the software program **112** to generate a document at a lower resolution which is calculated based on the difference between a standard document and the selected layout. Preferably, in a n-up

10   document, the resolution at which the software program **112** generates a document is defined as the user selected resolution divided by n.

Upon receiving the document generated a lower resolution than the selected resolution, the printer driver **114** makes a determination whether the printer controller **106** is capable of generating the selected layout and resolution from the lower resolution output of the software

15   program **112**. Upon a positive determination, the printer driver **114** forwards the output of the software program **112** to the printer controller **106**. The printer controller **106** then formats the document and generates the selected layout at the selected resolution. Upon a negative determination, the printer driver **114** generates the selected layout and resolution. After generating the selected layout and resolution, the printer driver **114** sends the formatted

20   document to the printer **104** for printing.

Although the preferred and alternate embodiments have been described in detail, it should be understood that various changes, substitutions and alterations can be made therein without departing from the spirit and scope of the invention as defined by the appended claims.